



An ellipsoid trust region bundle method for nonsmooth convex minimization

K.C. Kiwiel

► To cite this version:

K.C. Kiwiel. An ellipsoid trust region bundle method for nonsmooth convex minimization. RR-0586, INRIA. 1986. inria-00075968

HAL Id: inria-00075968

<https://inria.hal.science/inria-00075968>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tel: (1) 39 63 55 11

Rapports de Recherche

N° 586

**AN ELLIPSOID TRUST
REGION BUNDLE METHOD
FOR NONSMOOTH CONVEX
MINIMIZATION**

Krzysztof C. KIWIEL

Décembre 1986

AN ELLIPSOID TRUST REGION BUNDLE METHOD FOR NONSMOOTH CONVEX MINIMIZATION

UNE METHODE D'OPTIMISATION NON DIFFERENTIABLE COMBINANT FAISCEAUX ET ELLIPSOIDES

(*)
Krzysztof C. KIWIEL

Systems Research Institute, Polish Academy of Sciences
Newelska 6, 01-447 Varsovie, POLOGNE

RESUME

Cet article présente une méthode de faisceaux pour minimiser une fonction convexe f (éventuellement non différentiable) de plusieurs variables. A chaque itération, l'algorithme minimise un modèle linéaire par morceaux de f , soumis à une contrainte de région de confiance ellipsoïdale. La matrice de la contrainte quadratique, mise à jour à chaque itération comme dans la méthode des ellipsoïdes, a pour but de servir de "Hessien" tenant compte d'effets du "second ordre", permettant ainsi d'espérer une convergence plus rapide. On établit la convergence globale de la méthode et on donne des résultats numériques.

ABSTRACT

This paper presents a bundle method of descent for minimizing a convex (possibly nonsmooth) function f of several variables. At each iteration the algorithm finds a trial point by minimizing a polyhedral model of f subject to an ellipsoid trust region constraint. The quadratic matrix of the constraint, which is updated as in the ellipsoid method, is intended to serve as a generalized "Hessian" to account for "second-order" effects, thus enabling faster convergence. Global convergence of the method is established and numerical results are given.

AMS 1980 subject classification Primary : 65K05. Secondary : 90C25.

Key words : Nonsmooth Optimization, Nondifferentiable Programming, Convex Programming, Descent Method, Ellipsoid Algorithm.

(*) professeur invité dans le projet de M. LEMARECHAL du 12 au 23 mai 1986.



1 - INTRODUCTION

This paper presents a readily implementable algorithm for minimizing a convex (possibly nonsmooth) real-valued function f defined on \mathbb{R}^N . We suppose that the set of minimum points of f

$$X^* = \text{Arg min } f = \{x^* \in \mathbb{R}^N : f(x^*) \leq f(x) \forall x \in \mathbb{R}^N\}$$

is nonempty, and that we know center $x_c \in \mathbb{R}^N$ and radius $r > 0$ of some ball $E = \{x \in \mathbb{R}^N : |x - x_c| \leq r\}$ that intersects X^* . The algorithm only requires the computation of $f(x)$ and one arbitrary subgradient $g_p(x) \in \partial f(x)$ of f at each $x \in \mathbb{R}^N$.

The performance of the existing several methods for minimizing f depends on the shape of the level sets of f . The bundle methods of descent (see, e.g. [K2] for their survey), which can be derived by introducing a regularizing quadratic term in the cutting plane method [K1], seem to perform best when f is close to being piecewise linear (polyhedral) and the Haar condition holds at the minimum point. These methods are sensitive to the objective scaling (multiplication of f by a positive number), since so far no scale invariant rules are known for choosing the weights of their quadratic terms. On the other hand, the ellipsoid method (see [S1] and [Y1]), which can be described as a variable metric subgradient optimization method [G5], seems to work well when f has very elongated level sets, whereas its performance deteriorates for polyhedral functions with "fat" level sets (see [Y2, Section 9.5] and [G5]). This method is insensitive to the objective scaling.

The algorithm of this paper attempts to combine the best features of the ellipsoid and bundle methods. At each iteration it finds a trial point by minimizing a polyhedral model of f subject to an ellipsoid trust region constraint. The quadratic matrix of the constraint, which is updated as in the ellipsoid method, is intended to serve as a generalized "Hessian" to account for "second-order" effects, thus hopefully enabling faster convergence when the Haar condition does not hold. The trial point finding subproblem is solved approximately by estimating the Lagrange multiplier of its constraint and solving the resulting quadratic programming subproblem. In effect, the algorithm may also be viewed as a bundle method with an automatic choice of the quadratic

term and its weight, which is, in principle, insensitive to the objective scaling.

Our algorithm is intended for complex functions of relatively few variables, for which one function and subgradient evaluation dominates the effort per iteration involved in variable metric updates and quadratic programming subproblems. Hence it tries to minimize the number of function evaluations required to achieve a given accuracy. In particular, it evaluates f not at the center of the current ellipsoid, as do the existing ellipsoid methods (see [S2], [E1], [E2]), but at a point that should have a lower objective value according to the accumulated model of f .

Our interpretation of the generated ellipsoids as generalized "Hessians" is largely heuristic, since so far this notion has been made precise by Goffin [G4], [G7] only in the solution of linear inequalities. Alternative approaches to incorporating "second-order" models in the bundle methods are given in [L3], [L4] and [M1]. So far, they have not produced implementable algorithms.

We prove that the sequence of points generated by the method minimizes f . Rate of convergence results are still missing, but we report some encouraging numerical experience. Also it will be seen that there is much freedom in how the algorithm can be implemented, and we have only begun to explore some of the possibilities. In particular, we have been using simple ellipsoid cuts, whereas more refined ellipsoid updates would probably be more efficient.

We refer the reader to [A1] and [B1] for surveys and bibliographies of the ellipsoid method, and to [S3] for some of its modifications.

The paper is organized as follows. In Section 2 we derive the algorithm, which is stated in detail in Section 3. Its convergence is established in Section 4. Various more efficient ellipsoid updating strategies are discussed in Section 5. Section 6 describes an implementation of the method. Subgradient aggregation is introduced in Section 7. Numerical results are reported in Section 8. Finally, we have a conclusion Section.

We use the following notation. We denote by $\langle \cdot, \cdot \rangle$ and $|\cdot|$, respectively, the usual inner product and norm in finite-dimensional, real Euclidean space R^N . We use x_i to denote the i -th component of the vector x . Superscripts are used to denote different vectors, e.g. x^1 and x^2 . All vectors are column vectors. However, for convenience we sometimes write (x, y) for $(x^T, y^T)^T$, where T denotes transposition. For an $N \times N$ symmetric positive definite matrix A , we let A^{-1} denote the inverse of A , $A^{-T} = (A^{-1})^T$, $\langle x, y \rangle_A = \langle Ax, y \rangle = x^T A y$, $|x|_A = \langle Ax, x \rangle^{1/2}$. The volume $\text{Vol}(S)$ of a bounded measurable set S in R^N is its N dimensional Lebesgue measure.

For any $x \in R^N$,

$$\partial f(x) = \{ g \in R^N : f(y) \geq f(x) + \langle g, y-x \rangle \forall y \in R^N \}$$

denotes the subdifferential of f at x . The mapping $\partial f(\cdot)$ is locally bounded and f is continuous (see, e.g. [D1, Section 1.7.1 and Theorem 1.4.1]). $T(\alpha) = \{x \in R^N : f(x) \leq \alpha\}$ is the α -level set of f .

2 - DERIVATION OF THE METHOD

The algorithm to be described will generate two sequences of points $\{x^k\}_{k=1}^{\infty}$ and $\{y^k\}_{k=1}^{\infty}$ in R^N , where $x^1 = y^1$ is a given starting point. The sequence $\{x^k\}$ will satisfy $f(x^{k+1}) < f(x^k)$ if $x^{k+1} \neq x^k$, and $f(x^k) \rightarrow \min f$. The auxiliary trial points y^k will be used for computing $f(y^k)$ and $g^k = g_f(y^k)$ for all k . Also a sequence of ellipsoids

$$E_k = \{x \in R^N : |x - x_c^k|_{A_k} \leq 1\}$$

with centers $x_c^k \in R^N$ and symmetric positive definite matrices A_k will be generated such that $E_k \cap X^* \neq \emptyset$ for all k , where E_1 is a given starting ellipsoid.

At the k -th iteration, the algorithm will try to find a point y^{k+1} such that $f(y^{k+1}) < f(x^k)$. Ideally, y^{k+1} should minimize f . Since $E_k \cap X^* \neq \emptyset$, we may restrict the minimization to E_k , i.e. we may consider the subproblem

$$(2.1) \quad \text{minimize } f(y) \text{ over all } y \in E_k.$$

The trial point y^{k+1} will solve an approximate, but manageable version of subproblem (2.1), which is derived as follows.

The algorithm will use the following polyhedral approximation to f

$$\hat{f}^k(x) = \max \{f_j(x) : j \in J^k\} \text{ for all } x,$$

where J^k is a subset of $\{1, \dots, k\}$ and

$$f_j(x) = f(y^j) + \langle g_f(y^j), x - y^j \rangle \text{ for all } x$$

is the j -th linearization of f satisfying $f(x) \geq f_j(x)$ for all x . Replacing f by \hat{f}^k in (2.1), we obtain the subproblem

$$\text{minimize } \hat{f}^k(y) \text{ over all } y \in E_k$$

and its equivalent form

$$\text{minimize } u \text{ over all } (y, u) \in \mathbb{R}^N \times \mathbb{R}$$

$$(2.2) \quad \text{satisfying } f_j(y) \leq u \text{ for all } j \in J^k$$

$$\frac{1}{2} \|y - x_c^k\|_{A_k}^2 \leq \frac{1}{2}.$$

Let $\hat{\eta}^k$ denote the optimal Lagrange multiplier for the quadratic constraint of (2.2). It is not easy -and apparently not advisable- to solve (2.2) too accurately ; the algorithm will find (y^{k+1}, u^k) to

$$\text{minimize } \frac{1}{2} \eta^k \|y - x_c^k\|_{A_k}^2 + u \text{ over all } (y, u) \in \mathbb{R}^{N+1}$$

(2.3)

$$\text{satisfying } f_j(y) \leq u \text{ for all } j \in J^k$$

for some $\eta^k \geq \hat{\eta}^k$, so that $y^{k+1} \in E_k$. The search for η^k , which will test increasing values of η^k , will be similar to that employed in the trust region methods (see [M2] for a survey). Section 6 will suggest another argument in

favor of larger values of η^k .

Another motivation for subproblem (2.3) stems from the fact that, in a suitably transformed space, it reduces to the subproblems of the bundle methods of [K2, Chapter 2], thus inheriting their useful theoretical and computational properties. More specifically, the direction $\bar{d}^k = y^{k+1} - x_c^k$ and the predicted objective decrease

$$(2.4) \quad v^k = \hat{f}^k(y^{k+1}) - f(x^k) = u^k - f(x^k)$$

can be found by solving the subproblem

$$(2.5) \quad \begin{aligned} & \text{minimize } \frac{1}{2} \eta_k \|\bar{d}\|_{A_k}^2 + v \text{ over all } (\bar{d}, v) \in \mathbb{R}^{N+1} \\ & \text{satisfying } -\frac{\alpha_j^k}{\alpha_j} + \langle g^j, \bar{d} \rangle \leq v \text{ for } j \in J^k, \end{aligned}$$

where

$$(2.6) \quad \frac{\alpha_j^k}{\alpha_j} = f(x^k) - f_j(x_c^k) \text{ for } j \in J^k.$$

Let us represent the symmetric and positive definite matrices A_k and $B_k = A_k^{-1}$ as

$$A_k = \tilde{A}_k^T \tilde{A}_k \text{ and } B_k = \tilde{B}_k \tilde{B}_k^T$$

where $\tilde{B}_k = \tilde{A}_k^{-1}$ is a nonsingular $N \times N$ matrix (e.g. \tilde{B}_k^T is the Cholesky factor of B_k). Consider the space transformation

$$x \rightarrow \tilde{x} = \tilde{A}_k x,$$

which maps x^k , x_c^k , y^j , y^{k+1} and \bar{d}^k into \tilde{x}^k , \tilde{x}_c^k , \tilde{y}^j , \tilde{y}^{k+1} and \tilde{d}^k , respectively. In the transformed space (\tilde{x} -space), $\tilde{g}^j = \tilde{B}_k^T g_j$ are the subgradients of the function $\tilde{f}(\tilde{x}) = f(\tilde{A}_k^{-1} \tilde{x})$ at \tilde{y}^j , for $j \in J^k$, whereas (\tilde{d}^k, v^k) solves the transformed subproblem

$$(2.7) \quad \begin{aligned} & \text{minimize } \frac{1}{2} \eta^k \|\tilde{d}\|^2 + v \text{ over all } (\tilde{d}, v) \in \mathbb{R}^{N+1} \\ & \text{satisfying } -\bar{\alpha}_j^k + \langle \tilde{g}^j, \tilde{d} \rangle \leq v \text{ for } j \in J^k. \end{aligned}$$

The bundle methods of [K2, Chapter 2] use subproblem (2.7) for generating a descent direction for \tilde{f} at \tilde{x}_c^k . (In fact, they use $\eta^k = 1$ in (2.7), but η^k can be suppressed in (2.7) by replacing \tilde{A}_k with $(\eta^k)^{1/2} \tilde{A}_k$ in the transformation.) Therefore, by transforming "back" the results of [K2, pp.49 and 64], one can establish the following properties of subproblem (2.5). Let λ_j^k , $j \in J^k$, denote the (possibly nonunique) Lagrange multipliers of (2.5), and let

$$(2.8) \quad (p^k, \bar{\alpha}_p^k) \equiv \sum_{j \in J^k} \lambda_j^k (g^j, \bar{\alpha}_j^k)$$

and $\tilde{p}^k = \tilde{B}^T p^k$. Then $\tilde{d}^k = -\tilde{p}^k / \eta^k$, $v^k = \|\tilde{p}^k\|^2 / \eta^k + \bar{\alpha}_p^k$,

$$(2.9a) \quad \tilde{d}^k = -\frac{1}{\eta^k} B_k p^k,$$

$$(2.9b) \quad v^k = -\left\{ \frac{1}{\eta^k} \|p^k\|_{B_k}^2 + \bar{\alpha}_p^k \right\}.$$

Moreover,

$$(2.10) \quad w^k = \frac{1}{2\eta^k} \|p^k\|_{B_k}^2 + \bar{\alpha}_p^k$$

is the optimal value of the dual subproblem

$$(2.11) \quad \begin{aligned} & \text{minimize } \frac{1}{2\eta^k} \left\| \sum_{j \in J^k} \lambda_j^k \tilde{g}^j \right\|^2 + \sum_{j \in J^k} \lambda_j^k \bar{\alpha}_j^k, \\ & \text{subject to } \sum_{j \in J^k} \lambda_j^k = 1, \lambda_j^k \geq 0 \text{ for } j \in J^k, \end{aligned}$$

the solution set of which coincides with the set of Lagrange multipliers of (2.5). In particular,

$$(2.12) \quad \sum_{j \in J^k} \lambda_j^k = 1, \lambda_j^k \geq 0 \text{ for } j \in J^k.$$

In general, we would like the ellipsoid $E = E_k$ to be a tight approximation to (a portion of) X^* , since then the point $y^{k+1} \in E_k$ would be close to optimal.

Whenever we identify a redundant portion E^- of E such that $E^- \cap X^* = \emptyset$, we may reduce E by replacing it with a smaller ellipsoid E_+ that contains the remaining portion $E \setminus E^-$ as in the ellipsoid methods (see, e.g. [S2]). The ellipsoid methods update E as follows. Using one or more cutting planes, they choose a portion of E that contains $E \cap X^*$ and no more than half of E , and let E_+ be the least volume ellipsoid containing this portion. To ensure that at least half of E is cut off, so that $\text{Vol}(E_+) \leq q \text{Vol}(E)$ with $q < e^{-\frac{1}{2}(N+1)} < 1$ (see [T1]), they use the hyperplane $\{x \in \mathbb{R}^N : \langle g_f(x_c^k), x - x_c^k \rangle = 0\}$, or its translation (if $g_f(x_c^k) = 0$, they terminate with $x_c^k \in X^*$). Our algorithm will use the bounding hyperplane of

$$(2.13) \quad H_p^k = \{x \in \mathbb{R}^N : \langle p^k, x - x_c^k \rangle \leq \bar{\alpha}_p^k\}$$

defined via (2.8). This will save the computation of $g_f(x_c^k)$, but will provide a significant volume reduction only when $\bar{\alpha}_c^k \leq 0$. More specifically, if we let E_+ be the least volume ellipsoid containing $E \cap H_p^k$, then $\text{Vol}(E_+) \leq q \text{Vol}(E)$ if $\bar{\alpha}_p^k \leq 0$ and $p^k \neq 0$, whereas $E_+ \cap X^* \neq \emptyset$ if $(E \cap H_p^k) \cap X^* \neq \emptyset$, which will hold if

$$(2.14) \quad T(f(x^k)) \cap H_p^k,$$

because $E_k \cap X^* \neq \emptyset$. Now, multiplying by λ_j^k the relations

$$(2.15) \quad \begin{aligned} f(x) &\geq f(y^j) + \langle g_f(y^j), x - y^j \rangle = f_j(x) = \\ &= f(x^k) + \langle g^j, x - x_c^k \rangle - \bar{\alpha}_j^k \quad \text{for all } x \end{aligned}$$

and summing over $j \in J^k$, we deduce from (2.8) and (2.12) that

$$(2.16) \quad f(x) \geq f(x^k) + \langle p^k, x - x_c^k \rangle - \bar{\alpha}_p^k \quad \text{for all } x,$$

which establishes (2.14). Moreover, by (2.16), the algorithm may stop with $x^k \in X^*$ if $\bar{\alpha}_p^k \leq 0$ and $p^k = 0$. Thus we may replace E with E_+ to obtain the desired volume reduction if $\bar{\alpha}_p^k \leq 0$.

In geometric terms, the aggregate cut discussed above is provided by the aggregate linearization

$$\tilde{f}^k(x) = \sum_{j \in J^k} \lambda_j^k f_j(x) = \tilde{f}^k(x_c^k) + \langle p^k, x - x_c^k \rangle \quad \text{for all } x$$

with $\tilde{f}^k(x_c^k) = f(x^k) - \frac{-k}{\alpha_p}$; relation (2.16) means that $f(x) \geq \tilde{f}^k(x)$ for all x , and we have $H_p^k = \{x \in R^N : \tilde{f}^k(x) \leq f(x^k)\}$. A deep cut with $x_c^k \notin H_p^k$ is obtained if $\tilde{f}^k(x_c^k) > f(x^k)$ ($\frac{-k}{\alpha_p} < 0$). On the other hand, relation (2.15) shows that the "ordinary" linearizations f_j define cuts (called supercuts in [S2]) based on the relations

$$(2.17) \quad H_j^k = \{x \in R^n : \langle g^j, x - x_c^k \rangle \leq \frac{-k}{\alpha_j}\} = \{x : f_j(x) \leq f(x^k)\},$$

$$T(f(x^k)) \subset H_j^k.$$

Thus our aggregate cut is a convex combination of "ordinary" cuts (cf. (2.8) and (2.12)); it reduces to the surrogate cut of [G8] when f is polyhedral and $f(x^k) = \min f$. More about cuts will be said in Section 5.

A useful stopping criterion can be derived from inequality (2.16) as follows. Since $B_k = A_k^{-1}$ and the Cauchy-Schwarz inequality yields

$$\begin{aligned} |\langle p^k, x - x_c^k \rangle| &= |\langle B_k p^k, x - x_c^k \rangle_{A_k}| \leq \|B_k p^k\|_{A_k} \|x - x_c^k\|_{A_k} \\ &= \|p^k\|_{B_k} \|x - x_c^k\|_{A_k}, \end{aligned}$$

we may set $x = x^* \in E_k \cap X^*$ in (2.16) to obtain

$$(2.18) \quad f(x^k) \leq \min f + \|p^k\|_{B_k} + \frac{-k}{\alpha_p}.$$

Thus the algorithm may stop if $\|p^k\|_{B_k} + \frac{-k}{\alpha_p}$ is sufficiently small.

If neither an ellipsoid update nor termination occur, i.e. $\frac{-k}{\alpha_p} > 0$ and $p^k \neq 0$, then the predicted objective decrease v^k is negative (cf. (2.4) and (2.9b)). To ensure a significant objective reduction, the algorithm will take a serious step from x^k to $x^{k+1} = y^{k+1}$ only if

$$(2.19) \quad f(y^{k+1}) \leq f(x^k) + m v^k,$$

where $m \in (0,1)$ is a parameter. Otherwise, a null step with $x^{k+1} = x^k$ will occur, but the new linearization f_{k+1} of f at y^{k+1} will contribute to finding a better next trial point y^{k+2} .

3 - THE METHOD

We shall now state the simplest version of the method, postponing more efficient modifications till Sections 5 and 6.

Algorithm 3.1

Step 0 (Initialization). Choose a point $x_c^1 \in R^N$ and a symmetric positive definite $N \times N$ matrix A_1 such that the ellipsoid $E_1 = \{x \in R^N : |x - x_c^1|_{A_1} \leq 1\}$ satisfies $E_1 \cap X^* \neq \emptyset$. Select a starting point $x^1 \in R^N$, a final accuracy tolerance $\epsilon_s \geq 0$, a line search parameter $m \in (0,1)$ and weight updating parameters $\eta_u > 0$ and $\chi \in (1,100]$. Set $y^1 = x^1$ and $J^1 = \{1\}$. Compute $f(y^1)$, $g^1 = g_f(y^1)$ and $f_1(x_c^1) = f(y^1) + \langle g^1, x_c^1 - y^1 \rangle$. Choose $\eta^1 \in (0, \eta_u]$. Set the counters $k = 1$, $L = 0$ and $k(0) = 1$.

Step 1 (Direction finding). Find the solution (\bar{d}^k, v^k) and Lagrange multipliers λ_j^k , $j \in J^k$, of subproblem (2.5), with $\bar{\alpha}_j^k$, $j \in J^k$, given by (2.6). Compute p^k and $\bar{\alpha}_p^k$ by (2.8).

Step 2 (Stopping criterion). If $|p^k|_{B_k} + \bar{\alpha}_p^k \leq \epsilon_s$, terminate. If $\bar{\alpha}_p^k > 0$, go to Step 4; otherwise, continue.

Step 3 (Ellipsoid updating). Find $x_c^+ \in R^N$ and a symmetric positive definite $N \times N$ matrix A_+ such that $E_+ = \{x \in R^N : |x - x_c^+|_{A_+} \leq 1\}$ is the least volume ellipsoid containing $E_k \cap H_p^k$, where H_p^k is given by (2.13). Replace x_c^k and A_k by x_c^+ and A_+ , respectively, choose $\eta^+ \in (0, \eta_u]$, replace η^k by η^+ and go to Step 1.

Step 4 (Weight updating). If $|\bar{d}^k|_{A_k} \leq 1$, go to Step 5. Otherwise, choose $\eta^+ \in [\chi \eta^k, 100\eta^k]$, set $\eta^k = \eta^+$ and go to Step 1.

Step 5 (Line search). Set $y^{k+1} = x_c^k + \bar{d}^k$ and compute $f(y^{k+1})$ and $g^{k+1} = g_f(y^{k+1})$. If $f(y^{k+1}) \leq f(x^k) + m v^k$, set $x^{k+1} = y^{k+1}$, $k(L+1) = k+1$ and increase the counter of serious steps L by 1. Otherwise, set $x^{k+1} = x^k$.

Step 6 (Subgradient selection). Set $\hat{J}^k = \{j \in J^k : \lambda_j^k \neq 0\}$ and choose a set J^{k+1} satisfying $\hat{J}^k \subset J^{k+1} \subset J^k \cup \{k+1\}$.

Step 7. If $x^{k+1} = x^k$, set $\eta^{k+1} = \eta^k$; otherwise, choose $\eta^{k+1} \in (0, \eta_u]$. Set $x_c^{k+1} = x_c^k$ and $A_{k+1} = A_k$. Increase k by 1 and go to Step 1.

A few comments on the algorithm are in order.

Guidelines for choosing an initial ellipsoid can be found, for instance, in [G8] and [E2]. The obvious choice is to let $x_c^1 = x^1$ and $A_1 = (1/r)I$, where I is the identity matrix and $r > 0$ estimates the Euclidean distance from x^1 to X^* . It is reassuring to know that even if we had $E_1 \cap X^* = \emptyset$, the algorithm would still minimize f on E_1 , as will be proved in Section 4.

Step 1 can be implemented with the quadratic programming routine of [K3]; see Section 6.

Termination at Step 2 implies that $f(x^k) \leq \min f + \varepsilon_s$ (cf. (2.18)). (This estimate would be weakened to $f(x^k) \leq \min \{f(x) : x \in E_1\} + \varepsilon_s$ if we had $E_1 \cap X^* = \emptyset$; see Section 4.)

The ellipsoid update at Step 3 is well-defined, since the algebraic distance (in the metric defined by $|\cdot|_{A_k}$) from x_c^k to H_p^k

$$(3.1) \quad \omega_p^k = -\frac{\bar{\alpha}_p^k}{|\bar{p}^k|_{B_k}} = -\frac{\bar{\alpha}_p^k}{|\bar{p}^k|}$$

satisfies $\omega_p^k \in [0, 1)$ when $\bar{\alpha}_p^k \leq 0$ and $|\bar{p}^k| + \bar{\alpha}_p^k > 0$ after Step 2; note that in the \bar{x} -space $\tilde{E}_k = \tilde{A}_k E_k$ is the unit ball, whereas ω_p^k is the distance from \bar{x}_c^k to $\tilde{H}_p^k = \tilde{A}_k H_p^k = \{\bar{x} : \langle \bar{p}^k, \bar{x} - \bar{x}_c^k \rangle \leq \bar{\alpha}_p^k\}$. Hence one can compute x_c^{k+1} and $B_+ = A_+^{-1}$ as in [G8]; see Section 6. At the k -th iteration, an infinite number of returns to Step 1 from Steps 3 and 4 is possible only when $x^k \in X^*$ (see Section 4), an unlikely situation when f is not polyhedral.

The restrictions on the choice of the weighting coefficient η^k attempt to limit its growth. Specific choices of η^k will be discussed in Section 6. Here we may observe that if there is a cycle between Steps 1 and 4 without ellipsoid updates and one chooses $\eta^+ = \chi \eta^k$ at Step 4, then the cycle will terminate (see Section 4) with $\eta^k/\chi < \hat{\eta}^k \leq \eta^k$.

Observe that $x^k \in E_k$ for all k if $x^1 \in E_1$, since x^k is never cut off at Step 3 owing to (2.14), whereas $y^{k+1} \in E_k$ by construction, for all k .

The quadratic programming routine of [K3] will compute at most $N+1$ nonzero multipliers λ_j^k . Hence at Step 6 one can choose a set J^{k+1} with at most $N+2$ elements. This number of stored subgradients may be reduced by using subgradient aggregation ; see Section 7.

4 - CONVERGENCE

In this section we show that the algorithm minimizes f . Naturally, we assume that the final accuracy tolerance ϵ_s is set to zero.

We start with the ellipsoid updates.

Lemma 4.1. If Algorithm 3.1 did not stop before the k -th iteration, then at Step 1

$$(4.1) \quad T(f(x^k)) \in E_1 \cap E_k.$$

Proof. If $E_k = E_1$, the inclusion is obvious. Hence suppose that for some $j < k$ we have $E_1 \cap T(f(x^j)) \in E_j$ at Step 1 and a new ellipsoid E_+ is constructed at Step 3. Then $E_1 \cap T(f(x^j)) \in E_j \cap T(f(x^j)) \in E_j \cap H_p^k \cap E_+$ (cf. (2.14)). Since $f(x^{j+1}) \leq f(x^j)$, the desired conclusion follows by induction.

Since $E_1 \cap X^* \neq \emptyset$ by assumption, relation (4.1) implies that $E_k \cap X^* \neq \emptyset$. Hence we may use inequality (2.18) to obtain

Lemma 4.2. If Algorithm 3.1 terminates at the k -th iteration, then $x^k \in X^*$.

From now on we suppose that the algorithm does not terminate.

Owing to Lemma 4.1, the case of an infinite number of ellipsoid updates may be analyzed as in [G6].

Lemma 4.3. If Algorithm 3.1 executes Step 3 infinitely many times, then either k stays bounded and $x^k \in X^*$, or $f(x^k) \rightarrow \min f$ as $k \rightarrow \infty$.

Proof By construction, when Step 3 is entered with $\omega_p^k \in [0,1)$ (see (3.1)), E_+ satisfies $\text{Vol}(E_+) < e^{-\frac{1}{2}(N+1)} \text{Vol}(E_k)$ (see [T1]), and E_+ becomes E_k until the next update. Hence infinitely many updates lead to $\text{Vol}(E_k) \rightarrow 0$. Now, to derive a contradiction, suppose that there are infinitely many iterations with $f(x^k) \geq \bar{f}$, where $\bar{f} > f(x^*)$ for some $x^* \in E_1 \cap X^*$. Then $\text{Vol}(T(\bar{f}) \cap E_1) > 0$ from the continuity of f , whereas (4.1) yields $T(\bar{f}) \cap E_1 \subset E_k$. Thus $0 < \text{Vol}(T(\bar{f}) \cap E_1) \leq \text{Vol}(E_k)$ for all k , a contradiction with $\text{Vol}(E_k) \rightarrow 0$. Since $\{f(x^k)\}$ is nonincreasing, we deduce that $f(x^k) \rightarrow f(x^*)$ if $k \rightarrow \infty$. If Step 3 is executed infinitely many times for some fixed k , the same arguments show that $x^k \in X^*$.

From now on we suppose that only a finite number of ellipsoid updates occur. Then there exist $k_E \geq 1$ and an ellipsoid $\bar{E} = \{x \mid |x - \bar{x}_c|_{\bar{A}} \leq 1\}$ with center \bar{x}_c and a symmetric positive definite matrix \bar{A} such that after the last return from Step 3 to Step 1 that occurred at iteration k_E , if any, Step 1 is entered with

$$(4.2) \quad E_k = \bar{E}, \quad x_c^k = \bar{x}_c \quad \text{and} \quad A_k = \bar{A} \quad \text{if } k \geq k_E.$$

Let $\gamma > 0$ denote the square root of the minimum eigenvalue of \bar{A} , so that

$$(4.3) \quad \gamma |x| \leq |x|_{\bar{A}} \quad \text{for all } x.$$

We may now show that the algorithm cannot cycle infinitely between Steps 1 and 4 when its ellipsoid stays constant.

Lemma 4.4. Under the preceding assumptions, Algorithm 3.1 executes Step 5 at each iteration and there exists $\hat{\eta} > 0$ such that $\eta^k \leq \hat{\eta}$ for all k .

Proof. In view of the algorithms rules, it suffices to show that there exists $\tilde{\eta} > 100\eta_u$ such that if (4.2) holds, $k \geq k_E$ and $\eta \in (0, \tilde{\eta})$ then $|\bar{d}^k(\eta)| \frac{1}{A} \leq 1$, where $(\bar{d}^k(\eta), u^k(\eta))$

$$\text{minimize } \frac{1}{2}\eta |\bar{d}| \frac{2}{A} + u \quad \text{over all } (d, u) \in \mathbb{R}^{N+1}$$

$$\text{satisfying } f_j(\bar{x}_c) + \langle g^j, \bar{d} \rangle \leq u \quad \text{for } j \in J^k.$$

Since $\bar{d} = 0$ and $u = f(\bar{x}_c) \geq f_j(\bar{x}_c)$ for all $j \in J^k$ are feasible above,

$$(4.4) \quad \frac{1}{2}\eta |\bar{d}^k(\eta)| \frac{2}{A} + u^k(\eta) \leq f(\bar{x}_c).$$

Using the definition of f_j and the Cauchy-Schwarz inequality gives

$$\begin{aligned} u^k(\eta) &= \max \{f_j(\bar{x}_c) + \langle g^j, \bar{d}^k(\eta) \rangle : j \in J^k\} \geq \\ &\geq \min \{f(y^j) + \langle g^j, \bar{x}_c - y^j \rangle : j \in J^k\} - |\bar{d}^k(\eta)| \max_{j \leq k} |g^j|. \end{aligned}$$

Hence there exist constants $C_1 < f(\bar{x}_c)$ and $C_2 > 0$ such that

$$(4.5) \quad u^k(\eta) \geq C_1 - |\bar{d}^k(\eta)| C_2 \quad \text{for all } \eta > 0 \text{ and } k \geq k_E,$$

because $y^j \in E_j = \bar{E}$ if $j > k_E$, \bar{E} is bounded, f is continuous, $g^j \in \partial f(y^j)$ and ∂f is locally bounded. Combining (4.3) - (4.5), we get

$$f(\bar{x}_c) - C_1 \geq |\bar{d}^k(\eta)| \frac{1}{A} \left(\frac{1}{2}\eta |\bar{d}^k(\eta)| \frac{2}{A} - C_2/\gamma \right).$$

Therefore, if $|\bar{d}^k(\eta)| \frac{1}{A} > 1$ then $\eta/2 - C_2/\gamma \leq f(\bar{x}_c) - C_1$, and the existence of $\tilde{\eta}$ is clear.

By the rules of Step 5,

$$x^k = x^{k(L)} \quad \text{if } k(L) \leq k < k(L+1),$$

where for theoretical purposes we may let $k(L+1) = +\infty$ if the number L of serious steps stays bounded, i.e. if $x^k = x^{k(L)}$ for some fixed L and all $k \geq k(L)$. First we consider the case of unbounded L .

Lemma 4.5. Suppose that Algorithm 3.1 executes infinitely many serious steps. Then $x^k \rightarrow \bar{x}_c$ and $\bar{x}_c \in X^*$.

Proof. Let $K = \{k(L+1)-1 : L = 1, 2, \dots\}$, so that at Step 5 $f(x^{k+1}) \leq f(x^k) + m v^k$ for all $k \in K$. Since $-v^k = |p|_{B_k}^k / \eta^k + \bar{\alpha}_p^k > 0$ for $k \geq k_E$ and $m \in (0, 1)$ is fixed, whereas $f(x^k) \geq f(x^{k+1}) \geq \min f$ for all k , passing to the limit with $k \in K$ in the inequality $f(x^k) - f(x^{k+1}) \geq -m v^k$ yields $\bar{\alpha}_p^k \xrightarrow{K} 0$ and $|p|_{B_k}^k / \eta^k \xrightarrow{K} 0$. Letting $k \in K$ approach infinity in (2.18), we get $f(x^k) \rightarrow \min f$. By (4.2) and (2.9a), $\gamma |x^{k+1} - \bar{x}_c| \leq |x^{k+1} - \bar{x}_c|_{\bar{A}} = |y^{k+1} - \bar{x}_c|_{A_k} = |\bar{d}^k|_{A_k} = |p|_{B_k}^k / \eta^k$ for large $k \in K$, so $|p|_{B_k}^k / \eta^k \xrightarrow{K} 0$ implies $x^k \rightarrow \bar{x}_c$. Hence $f(x^k) \rightarrow f(\bar{x}_c)$ due to the continuity of f , and $f(\bar{x}_c) = \min f$, as desired.

It remains to consider the case of infinitely many successive null steps.

Lemma 4.6. Suppose that $x^k = x^{k(L)} = \bar{x}$ for some fixed L and all $k \geq k(L)$. Then $\bar{x} \in X^*$ and $\bar{x}_c \in X^*$.

Proof. In view of Lemma 4.4, the algorithm's rules imply the existence of $\bar{\eta} > 0$ and $\bar{k} > \max \{k(L), k_E\}$ such that $\eta^k = \bar{\eta}$ for all $k \geq \bar{k}$; otherwise, successive increases of η^k with $\chi > 1$ at Step 4 would make it unbounded, a contradiction.

Let $k \geq \bar{k}$ be fixed. Since $y^k = x_c^{k-1} + \bar{d}^{k-1}$, $x^k = x^{k-1} = \bar{x}$, $x_c^k = x_c^{k-1} = \bar{x}_c$ and $f(y^k) > f(x^{k-1}) + m v^{k-1}$, we obtain from (2.6)

$$\begin{aligned} \bar{\alpha}_k^k &= f(\bar{x}) - f(y^k) - \langle g^k, \bar{x}_c - y^k \rangle, \\ (4.6) \quad -\bar{\alpha}_k^k + \langle g^k, \bar{d}^{k-1} \rangle &> m v^{k-1}. \end{aligned}$$

Suppose the matrix $\bar{\eta} \bar{A}$ is factorized as $\bar{\eta} \bar{A} = \hat{A}^T \hat{A}$, where \hat{A} is $N \times N$ and nonsingular, and consider the space transformation

$$(4.7) \quad x \rightarrow \hat{x} = \hat{A}x,$$

which maps x^k , \bar{x} , y^j and \bar{d}^k into \hat{x}^k , \hat{x} , \hat{y}^j and \hat{d}^k , respectively. Also let $\hat{g}^j = \hat{A}^{-T} g^j$ and $\hat{\alpha}_j^k = \bar{\alpha}_j^k + \langle g^j, x_c^k - x^k \rangle$ for all $j \in J^k$, $\hat{p}^k = \hat{A}^{-T} p^k$ and

$$\bar{\alpha}_p^k = \frac{-k}{\alpha_p} + \langle p^k, x_c^k - x^k \rangle.$$

Thus for $k \geq \bar{k}$ we have $\eta^k A_k = \hat{A}^T \hat{A}$ and $\bar{\alpha}_p^k > 0$ ($k > k_E$), $\bar{\alpha}_k^k = \alpha_k^k + \langle g^k, \bar{x} - \bar{x}_c \rangle$ and, by (4.6),

$$- \frac{k}{\alpha_k} + \langle \hat{g}^k, \hat{d}^{k-1} \rangle > mv^{k-1}.$$

Hence one may use the various relations of Section 2 to deduce that for all $k \geq \bar{k}$ Algorithm 3.1 is essentially equivalent to the method of [K2, Section 2.5] applied to the convex function $\hat{f}(\cdot) = f(\hat{A}^{-1} \cdot)$ in the space transformed via (4.7). Then the results in [K2, Section 2.4 and 2.5] imply that $w^k = |p^k|_B^2 / 2\eta^k + \frac{-k}{\alpha_p} \rightarrow 0$ (see (2.10)). Therefore, $|p^k|_{B_k} \rightarrow 0$ and $\bar{\alpha}_p^k \rightarrow 0$, since $\bar{\alpha}_p^k > 0$ and $\eta^k = \eta$ for all $k > \bar{k}$, and (2.18) yields $\bar{x} \in X^*$.

It remains to show that $\bar{x}_c \in X^*$. Since $w^k \rightarrow 0$ and $\frac{-k}{\alpha_p} > 0$ for large k , (2.9b) and (2.10) imply that $v^k \rightarrow 0$. Hence $u^k = f(x^k) + v^k \rightarrow f(\bar{x})$. On the other hand, by (2.4),

$$(4.8) \quad u^k \geq f_k(y^{k+1}) = f(y^k) + \langle g_f(y^k), y^{k+1} - y^k \rangle$$

because $k \in J^k$ for all k . Since $|y^{k+1} - \bar{x}_c|_{\bar{A}} = |p^k|_{B_k} / \bar{\eta}$ for large k , and $|p^k|_{B_k} \rightarrow 0$, (4.3) implies that $y^k \rightarrow \bar{x}_c$. Passing to the limit in (4.8), we get $f(\bar{x}) \geq f(\bar{x}_c)$, since $u^k \rightarrow f(\bar{x})$, $y^k \rightarrow \bar{x}_c$, f is continuous and $g_f(\cdot)$ is locally bounded. Thus $\bar{x} \in X^*$ and $f(\bar{x}) \geq f(\bar{x}_c)$, so $\bar{x}_c \in X^*$; as desired.

We conclude from Lemmas 4.5 and 4.6 that the case of a finite number of ellipsoid updates is rather unlikely, since then the center of the last ellipsoid must be optimal.

Combining Lemmas 4.2 through 4.6, we deduce our principal result.

Theorem 4.7. Either the sequence $\{x^k\}$ generated by Algorithm 3.1 is finite and its last element minimizes f , or $\{x^k\}$ is infinite and $f(x^k) \rightarrow \min f$ as $k \rightarrow \infty$.

The key condition needed to ensure convergence is E_1 , $X^* \neq \emptyset$. Even if it fails, we still have

Theorem 4.8. If Algorithm 3.1 is applied to a general convex function f , which does not necessarily attain its infimum on R^N , then either of the following holds :

- (i) the sequence $\{x^k\}$ is finite and its last element x^k minimizes f on the set $\bigcup_{j=1}^k E_j$;
- (ii) $\{x^k\}$ is infinite and $\lim_{k \rightarrow \infty} f(x_k) \leq \inf \{f(x) : x \in \bigcup_{j=1}^{\infty} E_j\}$;
- (iii) $\{x^k\}$ is infinite and $\lim_{k \rightarrow \infty} f(x^k) = -\infty$.

Proof. Extend Lemma 4.1 by showing that

$$(4.9) \quad T(f(x^k)) \bigcup_{j=1}^k E_j \quad E_k$$

and use the proof of Lemma 4.3 with $\bar{f} > f(x)$ and an arbitrary $x \in \bigcup_{j=1}^k E_j$ to deduce that either (i) or (ii) holds if Step 3 is executed infinitely often. In the remaining case, suppose that $\{f(x^k)\}$ is bounded from below and obtain (ii) from the proofs of Lemmas 4.5 and 4.6.

Note that relations (2.16) and (4.9) imply that

$$f(x^k) \leq \min \{f(x) : x \in \bigcup_{j=1}^k E_j\} + |p^k|_{B_k} + \frac{-k}{\alpha_p},$$

which justifies the stopping criterion of Step 2 in the general case.

5 - ELLIPSOID UPDATING STRATEGIES

Proceeding as in [G6], one can establish an upper bound on the rate of convergence in objective values in terms of the rate of volume reduction of successive ellipsoids of the method. To obtain a faster volume reduction, the following modification will use more ellipsoid cuts.

Suppose that at Step 0 we choose a fixed $\bar{\omega} \in (-1/N, 0]$. Steps 2 and 3 are replaced by

Step 2' (Stopping criterion). If $|p^k|_{B_k} + \frac{\bar{\alpha}_p^k}{\alpha_p^k} \leq \epsilon_s$ or $|g^j|_{B_k} + \frac{\bar{\alpha}_j^k}{\alpha_j^k} \leq \epsilon_s$ for some $j \in J^k$, terminate ; otherwise, continue.

Step 3' (Ellipsoid updating). (i) Set the ellipsoid update indicator $i_E = 0$.

(ii) Compute the algebraic distances

$$\omega_p^k = -\frac{\bar{\alpha}_p^k}{\alpha_p^k} |p^k|_{B_k} \text{ and } \omega_j^k = -\frac{\bar{\alpha}_j^k}{\alpha_j^k} |g^j|_{B_k} \text{ for } j \in J^k$$

from x_c^k to H_p^k and H_j^k , respectively (cf. (2.13) and (2.17)). Let j^* maximize ω_j^k over $j \in J^k$. If $\omega_p^k \geq \omega_{j^*}^k$, set $\omega = \omega_p^k$ and $H = H_p^k$; otherwise, set $\omega = \omega_{j^*}^k$ and $H = H_{j^*}^k$.

(iii) If $\omega < \bar{\omega}$, go to (v) ; otherwise, continue.

(iv) Let $E_+ = \{x \in \mathbb{R}^N : |x - x_c^+|_{A_+} \leq 1\}$ be the least volume ellipsoid containing $E_k \cap H$. Set $i_E = 1$, replace x_c^k and A_k by x_c^+ and A_+ , and go to (ii).

(v) If $i_E = 0$, go to Step 4 ; otherwise, choose $\eta^+ \in (0, \eta_u]$, replace η^k by η^+ and go to Step 1.

The above modification uses "ordinary" cuts based on relations (2.17), which provide the following analogue of (2.18)

$$(5.1) \quad f(x^k) \leq \min f + |g^j|_{B_k} + \frac{\bar{\alpha}_j^k}{\alpha_j^k}$$

for the stopping criterion. Step 3'(ii) chooses the (possibly nonunique) deepest cut if $\omega > 0$, or the least shallow cut if $\omega \leq 0$. The condition $\omega \geq \bar{\omega} > -1/N$ ensures a significant volume reduction, since at Step 3'(iv) we have $\text{Vol}(E_+) = q(\omega) \text{Vol}(E_k)$ with

$$q(\omega) = \left(\frac{N^2}{N^2-1}\right)^{(N-1)/2} \frac{N}{N+1} (1-\omega^2)^{(N-1)/2} (1-\omega),$$

$$q(\omega) \leq q(\bar{\omega}) \leq \bar{q} := e^{-N(\bar{\omega}-1/N)/3} < 1$$

for $N > 1$ (see [T1], [G7]). On the other hand, $\omega < \bar{\omega}$ implies that

$$(5.2) \quad |\bar{\omega}| E_k \subset T^k \subset E_k,$$

where $T^k = E_k \cap \bigcap_{j \in J^k} H_j^k$ is an outer approximation to $E_k \cap T(f(x^k))$. If we had $\bar{\omega} = -1/N$ and $T^k = T(f(x^k))$, relation (5.2) would mean that the matrix A_k of E_k is a generalized "Hessian" of [G7]. (This terminology comes from the fact that for a smooth convex f with a minimizer x^* the classical Hessian at x^* is associated with the limit of the smallest ellipsoid containing (or the largest ellipsoid contained in) $[T(f(x^*)) + \varepsilon^2/2] - x^*/\varepsilon$ as $\varepsilon \rightarrow 0$; see [G7] for details.) Wanting E_k to be close to T^k , and hence to $T(f(x^k))$, we would like to have a large value of $|\bar{\omega}|$ in (5.2). Since such a value cannot, in general, be greater than $1/N$ (see [G7]), whereas $\bar{\omega}$ close to $-1/N$ may result in exceedingly many updates, in practice we use $\bar{\omega} = -1/2N$.

The deepest aggregate (or surrogate) cut is defined if x_c^k is cut off by at least one half-space H_j^k with $\bar{\alpha}_j^k > 0$. This cut is given by the half-space \hat{H}^k containing $\bigcap_{j \in J^k} H_j^k$ that is furthest from x_c^k in the metric of $|\cdot|_{A_k}$. \hat{H}^k can be found by projecting x_c^k on $\bigcap_j H_j^k$. Thus we may find d_H^k to

$$(5.3) \quad \begin{cases} \text{minimize } \frac{1}{2} \|d\|_{A_k}^2, \\ \text{subject to } -\frac{\bar{\alpha}_j^k}{\alpha_j^k} + \langle g^j, d \rangle \leq 0 \text{ for } j \in J^k, \end{cases}$$

since $\hat{H}^k = \{x : \langle -A_k d_H^k, x - x_c^k \rangle \leq -\|d_H^k\|_{A_k}^2\}$. Equivalently, we may find the Lagrange multipliers $\lambda_{H,j}^k$ of (5.3) that

$$\begin{cases} \text{minimize } \frac{1}{2} \left\| \sum_{j \in J^k} \lambda_j^k g^j \right\|_{B_k}^2 + \sum_{j \in J^k} \bar{\alpha}_j^k \lambda_j^k, \\ \text{subject to } \lambda_j^k \geq 0 \text{ for } j \in J^k, \end{cases}$$

and let

$$(p_H^k, \alpha_H^k) = \sum_{j \in J^k} \lambda_{H,j}^k (g^j, \bar{\alpha}_j^k),$$

since then $d_H^k = -B_k p_H^k$ and

$$(5.4) \quad \hat{H}^k = \{x \in R^N : \langle p_H^k, x - x_C^k \rangle \leq \alpha_H^k\}.$$

Of course, finding the deepest aggregate cut involves additional work in quadratic programming. Therefore, it is worthwhile to observe that the usual aggregate cut may be close to the deepest one. For instance, $\hat{H}^k = H_p^k$ if $v^k = 0$. Indeed, in this case subproblems (2.5) and (5.3) produce $\bar{d}^k = d_H^k$, $p^k = p_H^k$ and $\bar{\alpha}_p^k = \alpha_H^k$, so that relations (2.13) and (5.4) imply that $H_p^k = \hat{H}^k$. This suggests that H_p^k approximates \hat{H}^k whenever $|v^k|$ is small relative to $|g^j|_{B_k}$ for $j \in J^k$.

There exists an additional possibility for updating the ellipsoid after a serious step. When $x^{k+1} \neq x^k$ at Step 7, we may compute $\bar{\alpha}_j^{k+1}$, $j \in J^k$, $\bar{\alpha}_p^{k+1} = \bar{\alpha}_p^k + f(x^{k+1}) - f(x^k)$, increase k by 1 and, before going to Step 1, execute Steps 2' and 3'.

So far we have restricted our discussion to single cuts. To obtain a greater reduction in volume, at Step 3'(iv) we may choose a subset J_H^k of J^k such that $\omega_j^k \geq \bar{\omega}$ for some $j \in J_H^k$, and then let E_+ be the least volume ellipsoid containing the set $S^k = E_k \cap \bigcap_{j \in J_H^k} H_j^k$. In practice the construction of E_+ may be too complicated if J_H^k has more than two elements (see [S3]); for two elements explicit formulae are given in [G1], [E1]. In fact, it is not absolutely necessary to use minimum volume ellipsoids, and we may let E_+ be any ellipsoid containing S^k such that $\text{Vol}(E_+) \leq q(\bar{\omega}) \text{Vol}(E_k)$. This extra freedom may facilitate the construction of E_+ .

It is straightforward to verify that all the modifications discussed in this section are covered by the convergence analysis of Section 4.

We conclude that the algorithm can use a variety of techniques for updating its ellipsoids. Naturally, the best strategy is open to question.

6 - IMPLEMENTATION

In this section we discuss our implementation of the algorithm.

As in [G8], at the k -th iteration we use the factorization

$$B_k = LDL^T$$

with L a lower triangular $N \times N$ matrix with unit diagonal and $D = \text{diag}(d_1, \dots, d_N)$ a diagonal matrix with positive diagonal. Let $\tilde{B}_k = LD^{\frac{1}{2}}$, so that $B_k = \tilde{B}_k \tilde{B}_k^T$. We do not store the subgradients g^j , but update the vectors

$$(6.1) \quad \tilde{g}^j = \tilde{B}_k^T g^j = D^{\frac{1}{2}} L^T g^j \quad \text{for } j \in J^k.$$

Step 1 is implemented with the quadratic programming routine of [K3]. This routine solves the following version of the dual subproblem (2.11)

$$(6.2) \quad \begin{aligned} & \text{minimize } \frac{1}{2} \left| \sum_{j \in J^k} \lambda_j \tilde{g}^j \right|^2 + \sum_{j \in J^k} \lambda_j \eta^k \frac{\bar{\alpha}_j^k}{\alpha_j^k}, \\ & \text{subject to } \sum_{j \in J^k} \lambda_j = 1, \lambda_j \geq 0 \text{ for } j \in J^k. \end{aligned}$$

It also computes, as by-products, the quantities

$$(6.3) \quad (\bar{p}^k, \bar{\alpha}_p^k) = \sum_{j \in J^k} \lambda_j^k (\tilde{g}^j, \bar{\alpha}_j^k),$$

$|p^k|_{B_k}^2 = |\bar{p}^k|^2$ and $\eta^k v^k = -\{|\bar{p}^k|^2 + \eta^k \bar{\alpha}_p^k\}$ (cf. (2.9b)). Hence we can calculate $|d^k|_{A_k}^2 = |\bar{p}^k|^2 / (\eta^k)^2$ and

$$(6.4) \quad \bar{d}^k = - (1/\eta^k) \tilde{B}_k \bar{p}^k = - (1/\eta^k) LD^{\frac{1}{2}} \bar{p}^k.$$

The algorithm terminates at Step 2' if

$$(6.5) \quad \min \{ |\bar{p}^k| + \bar{\alpha}_p^k, |\tilde{g}^j| + \bar{\alpha}_j^k : j \in J^k \} \leq \epsilon_s (1 + |f(x^k)|),$$

where $\epsilon_s > 0$ is the desired relative final accuracy in the objective value, i.e. $f(x^k) - \min f \leq \epsilon_s (1 + |f(x^k)|)$ at termination; see (2.18) and (5.1).

(This stopping criterion is less sensitive to problem scaling than that of Algorithm 3.1.) Hence we compute the norms $\|g^j\|$, which are also needed by the quadratic programming routine.

For simplicity, we use the single cuts described at Step 3' in Section 5.

The ellipsoid updating at Step 3'(iv) is trivial if $N = 1$. Hence suppose that $N > 1$ and H_p^k is used to cut E_k . As in [G8], we compute $\omega_p^k = -\bar{\alpha}_p^k / \|\bar{p}^k\|$,

$$(6.6) \quad \left\{ \begin{array}{l} \omega = \min \{ \omega_p^k, 1 - \epsilon_M \}, \\ t_c = (1 + N\omega) / (N+1) \|\bar{p}^k\|, \\ x_c^+ = x_c - t_c L D^{\frac{1}{2}} \bar{p}^k, \\ \delta = N^2 (1 - \omega^2) / (N^2 - 1), \\ \sigma = 2(1 + N\omega) / (N+1) (1 + \omega), \end{array} \right.$$

where the use of the relative machine precision ϵ_M ensures that the updated matrix $B_+ = A_+^{-1}$ given by

$$B_+ = \delta [B_k - \sigma B_k p^k (B_k p^k)^T / \|p^k\|_{B_k}^2]$$

can be factorized as $B_+ = L_+ D_+ L_+^T$ with L_+ lower triangular and D_+ diagonal with positive diagonal. More specifically, we compute

$$\gamma = D^{\frac{1}{2}} \bar{p}^k / \|\bar{p}^k\|,$$

so that $B_+ = L [\delta(D - \sigma \gamma \gamma^T)] L^T$, and then find a diagonal matrix $\tilde{D} = \text{diag}(\tilde{d}_1, \dots, \tilde{d}_N)$ and a unit lower triangular matrix $\tilde{L} = (\tilde{L}_{ij})$ such that $D - \sigma \gamma \gamma^T = \tilde{L} \tilde{D} \tilde{L}^T$ by using the recurrence relations

$$(i) \quad \text{set } t_{N+1} = \frac{N-1}{N+1} \frac{1-\omega}{1+\omega};$$

$$(ii) \quad \text{for } j = N-1, \dots, 1 \text{ set}$$

$$t_j = t_{j+1} + \sigma \gamma_j^2 / d_j,$$

$$\tilde{d}_j = d_j t_{j+1} / t_j,$$

$$\beta_j = -\sigma \gamma_j / (d_j t_{j+1})$$

and setting $\tilde{l}_{ij} = \gamma_i \beta_j$ for $j < i$ (see [G3]). Then $D_+ = \delta \tilde{D}$ and the product $L_+ = (l_{ij}^+)$ can be computed from the recurrence relations (see [G2]) :

$$(i) \quad \text{set } q_N^N = \gamma_N,$$

$$(ii) \quad \text{for } j = N-1, \dots, 1 \text{ set}$$

$$q_j^j = \gamma_j,$$

$$\left. \begin{aligned} l_{ij}^+ &= l_{ij} + \beta_j q_i^{j+1} \\ q_i^j &= q_i^{j+1} + \gamma_j l_{ij} \end{aligned} \right\} \text{ for } i = j+1, \dots, N.$$

To update the quantities $\bar{\alpha}_j^k = f(x^k) - f_j(x_c^k)$ we use the relations

$$f_j(x_c^+) - f_j(x_c^k) = \langle g^j, x_c^+ - x_c^k \rangle = \langle \bar{g}^j, \tilde{B}_k^{-1} (x_c^+ - x_c^k) \rangle = -t_c \langle \bar{g}^j, \bar{p}^k \rangle,$$

which follow from (6.1) and (6.6). Next, to update the transformed subgradients (6.1), we note that

$$D_+^{\frac{1}{2}} L_+^T g^j = D_+^{\frac{1}{2}} \tilde{L}^T L^T g^j = D_+^{\frac{1}{2}} \tilde{L}^T D^{-\frac{1}{2}} \bar{g}^j,$$

so that $\bar{g}^j = D_+^{\frac{1}{2}} L_+^T g^j$ can be computed from the backward recurrence (see [G2]) :

$$(i) \quad \text{set } s = 0, \bar{g}_N^j = \tilde{d}_N \bar{g}_N^j;$$

$$(ii) \quad \text{for } i = N, N-1, \dots, 2 \text{ set}$$

$$s = s + \tilde{\gamma}_i \bar{g}_i^j,$$

$$\bar{g}_{i-1}^j = \tilde{d}_{i-1} \bar{g}_{i-1}^j + \tilde{\beta}_{i-1} s,$$

where the quantities $\tilde{d}_i = (d_i^+/d_i)^{1/2}$, $\tilde{\gamma}_i = \gamma_i / d_i^{1/2}$, $\tilde{\beta}_i = \beta_i (d_i^+)^{1/2}$ are computed beforehand, using the stored values of $d_i^{1/2}$ and $(d_i^+)^{1/2}$.

The convergence analysis of Section 4 imposes only weak restrictions on the choice of $\{\eta^k\}$. Relation (5.2) with $|\bar{\omega}| = 1/2N$ suggests that it may be useful to restrict the trial point finding to the smaller ellipsoid $|\bar{\omega}| E_k$ which, being an inner approximation to T^k , should be almost contained in $T(f(x^k))$. Hence our strategy for selecting η^k aims for y^{k+1} to

$$\text{minimize } \hat{f}^k(y), \text{ subject to } y \in r_e E_k$$

or equivalently for $\bar{d}^k = y^{k+1} - x_c^k$ to

$$(6.7) \quad \text{minimize } \hat{f}^k(x_c^k + \bar{d}), \text{ subject to } |\bar{d}|_{A_k} \leq r_e,$$

where $r_e \in (0, 1]$ is a trust region radius. We use the fixed value $r_e = 1/2N$ (corresponding to $\bar{\omega} = -1/2N$), which seems to work better than $r_e = 1$, although an adaptive choice of r_e at each iteration could be more efficient.

We relate subproblems (6.2) and (6.7) through the following choice of η^k . At Steps 3'(v) and 7 we set η^+ and η^{k+1} (if $x^{k+1} \neq x^k$) to the value of $\eta_{\min} = 10^{-5}$, since smaller values may lead to inaccuracies at quadratic programming (cf. (6.4)). At Step 4, if $|\bar{d}^k|_{A_k} > 1.2 r_e$, η^k is increased to

$$(6.8) \quad \eta^+ = \max \{1.2 |\bar{d}^k|_{A_k} / r_e, \chi\} \eta^k,$$

where $\chi > 1$ is a parameter. When η^+ replaces η^k in (6.2), the corresponding \bar{p}^+ and \bar{d}^+ given by (6.3) and (6.4) satisfy $|\bar{p}^+| \geq |\bar{p}^k|$, $|\bar{d}^+|_{A_k} = |\bar{p}^+| / \eta^+$ and

$$|\bar{d}|_{A_k} \geq |\bar{d}^k|_{A_k} \eta^k / \eta^+ \geq \min \{1/1.2, 1.2/\chi\} r_e.$$

Hence eventually we get

$$(6.9) \quad \min \{1/1.2, 1.2/\chi\} r_e \leq |\bar{d}^k|_{A_k} \leq 1.2 r_e,$$

which implies satisfaction of the trust region constraint of (6.7) with the relative accuracy of 20% if $\chi = 1.5$. It might appear that, since η^k is not decreased after null steps with no ellipsoid updates, one could have $|\bar{d}^k|_{A_k}$ much smaller than r_e at Step 4. Yet in our calculations the bound (6.9) was (slightly) violated in negligibly few cases.

One could, of course, consider other ways of selecting η^k , e.g. safeguarded interpolation using parametric analysis of subproblem (6.2) with respect to η^k . We note that when the change of η^k is small enough, our quadratic programming routine [K3] can solve the new subproblem very quickly by exploiting the information gathered so far.

We use the following subgradient selection strategy. At Step 0 we choose the maximum number $M_g \geq N+2$ of stored subgradients. At Step 6 we initially set $J^{k+1} = J^k \setminus \{k+1\}$ and then, if necessary, drop from J^{k+1} an index $j \in \hat{J}^k$ with the largest value of $f(x^{k+1}) - f_j(x_c^k)$, so that J^{k+1} has at most M_g elements.

It is worth adding that, in theory, the algorithm can be made invariant with respect to the objective scaling. To this end, consider the following version. At Step 0 we choose $r_e \in (0,1]$ and set $\eta^1 = |\bar{p}^1| / r_e$. At Step 2' we delete the '1' in the stopping criterion (6.5). At Steps 3'(v) and 7 we set η^+ and η^{k+1} (if $x^{k+1} \neq x^k$) equal to $\eta_u \eta^k$. At Step 4 η^k is not changed if $|\bar{d}^k|_{A_k} \leq 1.2 r_e$; otherwise, η^+ is calculated from (6.8). Moreover, suppose that the arbitrary decisions of Steps 3'(ii) and 6, concerning the selection of cuts and subgradients, are made according to some fixed rules that account for the possible ties. Note that the proof of Lemma 4.4 can be extended to cover this version.

The above-described version is scale invariant in the following sense. Applying the algorithm to f , we get sequences $\{x^k\}$, $\{\bar{d}^k\}$, $\{A_k\}$, $\{\eta^k\}$, etc. Next,

suppose we use the same parameters at Step 0 and apply the algorithm to sf , where $s > 0$ is fixed, with the subgradient mapping sg_f to get sequences $\{x_s^k\}$, $\{\bar{d}_s^k\}$, $\{A_s^k\}$, $\{\eta_s^k\}$, etc. Then $x_s^k = x^k$, $\bar{d}_s^k = \bar{d}^k$, $A_s^k = A_k$ and $\eta_s^k = s \eta^k$ for all k such that termination does not occur before iteration k . To save space, we omit an inductive proof of this fact. (Some hints for the proof : multiply the objective of (6.2) by s^2 and relations (6.3), (6.8), (2.4) and (2.19) by s , observe that the division in (6.4) cancels s , and use the uniqueness of $(p^k, \bar{\alpha}_p^k)$ and relations (2.13) and (2.17) for ellipsoid updates.)

7 - SUBGRADIENT AGGREGATION

The algorithm described so far will typically use $N+2$ past subgradients at each iteration. We shall now show how the subgradient aggregation strategy of [K2] can be used to trade-off storage and work per iteration for speed of convergence.

At Step 0 we choose the maximum number $M_g \geq 1$ of stored "ordinary" subgradients. The aggregate subgradient is a convex combination of "ordinary" subgradients, which is generated recursively as follows. At Step 0 we define $p^0 = g^1$ and $\tilde{f}^0(x) = f(y^1) + \langle p^0, x - x^1 \rangle$ for all x . At Step 1 we append to subproblem (2.5) the aggregate constraint

$$(7.1) \quad -\frac{\bar{\alpha}_p^{k-1}}{\alpha_p^{k-1}} + \langle p^{k-1}, \bar{d} \rangle \leq v,$$

where $\frac{\bar{\alpha}_p^{k-1}}{\alpha_p^{k-1}} = f(x^k) - \tilde{f}_p^{k-1}(x_c^k)$, and use the Lagrange multiplier λ_p^k of (7.1) to define

$$(p^k, \bar{\alpha}_p^k) = \sum_{j \in J^k} \lambda_j^k (g^j, \bar{\alpha}_j^k) + \lambda_p^k (p^{k-1}, \bar{\alpha}_p^{k-1}).$$

As before, we define $\tilde{f}^k(x) = f(x^k) - \frac{\bar{\alpha}_p^k}{\alpha_p^k} + \langle p^k, x - x_c^k \rangle$ to close the recursion. Then at Step 6 we may let J^{k+1} contain $k+1$ as well as any other $M_g - 1$ past indices, e.g. indices j with the largest values of ω_j^k .

Reasoning as in [K2, Section 2.4], one may verify that the convergence results of Section 4 remain valid for the version with subgradient aggregation.

8 - NUMERICAL EXAMPLES

We shall now report on computational testing of the algorithm using a double precision Fortran code on an IBM PC/XT clone microcomputer with relative accuracy $\epsilon_m = 2.2 \times 10^{-16}$ ($= 2.2E-16$).

The parameters of the algorithm had the values $m = 0.1$, $\eta_u = 1E-5$, $\chi = 2$, $\bar{\omega} = -1/2N$ (cf. (5.2)) and $r_e = 1/2N$ (cf. (6.7)).

Table 8.1 summarizes results for several standard test problems taken from the literature, which are reviewed below. The following notation is used. N is the number of variables, M_g is the maximum number of stored past subgradients, x^* is the known solution of a problem, x^1 is the standard starting point, $V^1 = \text{Vol}(E_1)$, k is the iteration number at termination, $V^k = \text{Vol}(E_k)$ is the volume of the final ellipsoid, and m_e is the total number of ellipsoid updates. The stopping criterion (6.5) was used with a value of ϵ_s chosen for each problem so as to facilitate comparison with results reported in the literature. For each run of the algorithm, an initial ellipsoid was specified by choosing positive tolerances δ_i such that $|x_i^1 - x_i^*| \leq \delta_i$ and setting $d_i = N \delta_i^2$ for $i = 1, \dots, N$ in the initial matrix $B_1 = \text{diag}(d_1, \dots, d_N)$. Typically, three runs with increasing δ_i (denoted by δ^a , δ^b and δ^c) are reported for each problem.

Example 8.1 The Shor problem has $N = 5$,

$$f(x) = \max \{ b_i - \sum_{j=1}^5 (a_{ij} - x_j)^2 : i = 1, \dots, 10 \},$$

$$x^* = (1.12434, 0.97945, 1.47770, 0.92023, 1.12429),$$

$x^1 = (0, 0, 0, 0, 1)$, $\delta_i^{1,a} = |x_i^1 - x_i^*|$ for all i , $\delta_i^{1,b} = 2$ for all i , $\delta_i^{1,c} = 10$ for all i (see [S1, p.137] for the data a_{ij} and b_i).

Example 8.2 The first Colville problem has $N = 5$,

$$f(x) = \sum_{j=1}^5 d_j x_j^3 + \sum_{i=1}^5 \sum_{j=1}^5 c_{ij} x_i x_j + \sum_{j=1}^5 e_j x_j + 50 \max \{ F(x), 0 \},$$

$$F(x) = \max \{ b_i - \sum_{j=1}^5 a_{ij} x_j : i = 1, \dots, 10 \},$$

$x^* = (0.3, 0.3335, 0.4, 0.4285, 0.224)$, $x^1 = (0,0,0,0,1)$, $\delta_i^{2,a} = |x_i^1 - x_i^*|$, $\delta_i^{2,b} = 1$ and $\delta_i^{2,c} = 10$ for all i (see, e.g. [K2, p.350] for the problem data).

Example 8.3 The Rosen-Suzuki problem has $N = 4$,

$$f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 + F(x),$$

$$F(x) = 5 \max \{F_1(x), F_2(x), F_3(x), 0\},$$

$$F_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8,$$

$$F_2(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10,$$

$$F_3(x) = x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5,$$

$x^* = (0,1,2,-1)$, $x_1 = (0,0,0,0)$, $\delta_i^{3,a} = (1 \text{ E-}4, 1, 2, 1)$, $\delta_i^{3,b} = 3$ and $\delta_i^{3,c} = 10$ for all i .

Example 8.4 Lemarechal's MAXQUAD problem has $N = 10$,

$$f(x) = \max \{ \langle A^L x, x \rangle - \langle b^L, x \rangle : L = 1, \dots, 5 \},$$

$$A_{ij}^L = A_{ji}^L = \exp(i/j) \cos(i \cdot j) \sin(L), \quad i \neq j,$$

$$A_{ii}^L = i |\sin(i)| / 10 + \sum_{j \neq i} |A_{ij}^L|,$$

$$b_i^L = \exp(i/L) \sin(i \cdot L).$$

(This problem is difficult to quote : misprints abound in [K2, pp.346-347], [L2, p.151], [Z1, Example 7.2], etc.) In [L2, p.152] the optimum value $f(\bar{x}) = -0.8414$ is quoted for a point \bar{x} which in fact has $f(\bar{x}) = -0.8411$. The best point known to us is

$$x^* = (-0.126257, -3.43783\text{E-}2, -6.85716\text{E-}3, 2.63606\text{E-}2, 6.72949\text{E-}2)$$

$$(-0.278400, 7.42187\text{E-}2, 0.138524, 8.40313\text{E-}2, 3.85804\text{E-}2)$$

with $f(x^*) = -0.841408$. We used $x^1 = 0$, $\delta_i^{4,a} = |x_i^*|$, $\delta_i^{4,b} = 0.3162$ (corresponding to $d_i = 1$) and $\delta_i^{4,c} = 2$ for all i .

Example 8.5 This academic problem MXHILB with

$$f(x) = \max \left\{ \left| \sum_{j=1}^N x_j / (i+j-1) \right| : i = 1, \dots, N \right\}, x^* = 0,$$

corresponds to solving the equation $Ax = b$, where $b = 0$ and A is an $N \times N$ section of the Hilbert matrix. We used $x^1 = (1, \dots, 1)$ and $\delta_i^{5,a} = 5$ for all i with $N = 30$ and $N = 50$.

Example 8.6 Problem L1HILB with

$$f(x) = \sum_{j=1}^N \left| \sum_{i=1}^N x_i / (i+j-1) \right|, x^* = 0,$$

is a more difficult version of MXHILB, since it has 2^N linear pieces, whereas MXHILB has $2N$. We used $x_i^1 = 1$, $\delta_i^{6,a} = 5$ for $i = 1, \dots, N$, $N = 30, 50$.

Example 8.7 Lemarechal's SHELL DUAL problem [L2, p.154] has a highly nonconvex objective function f of 15 variables, with $f(x^*) = 32.3488$ for

$$x^* = (0.3, 0.3335, 0.4, 0.4283, 0.224, 0, 0, 5.1741,$$

$$0, 3.0611, 11.8396, 0, 0, 0.1039, 0).$$

This problem seems to be very difficult for general-purpose descent methods (see [L1]). In order to tackle nonconvexity, we incorporated in the algorithm the two-point line search of [K2, p.103] (with parameters $\bar{t} = 0.01$, $\xi = 0.1$, $\gamma = 10$). To this end, we computed the search direction $d^k = \bar{d}^k + x_c^k - x^k$ from x^k , and the line search procedure found two stepsizes t_L^k and t_R^k , $0 \leq t_L^k \leq t_R^k \leq 1$, the next iterate $x^{k+1} = x^k + t_L^k d^k$ and the next trial point $y^{k+1} = x^k + t_R^k d^k$ which provided the next linearization f_{k+1} of f . Of course, the algorithm is not guaranteed to minimize a nonconvex f . Nevertheless, for $x_i^1 = 1E-4$, $i \neq 12$, $x_{12}^1 = 60$ and

$$\delta = (0.3, 0.3335, 0.4, 0.4283, 0.224, 0.1, 0.1, 10, \\ 0.1, 5, 15, 100, 0.1, 1, 0.1)$$

$(f(x^1) = 2400, \|x^1 - x^*\|_{A_1} = 0.66, V^1 = 7.3E+5)$, the algorithm did converge to an approximate solution, although the usual stopping criterion (6.5) did not work with $\epsilon_s = 1E-4$ and termination occurred due to exceeding the function evaluation limit of 600. Table 8.2 illustrates the algorithm's progress, with NFEV denoting the total number of function and subgradient evaluations.

Our results for problems 8.1-8.4 and 8.7 may be compared with those in [S1, p.139], [L1], [Z1] and [K2, pp.346-348]. Of course, no conclusions should be drawn from such limited computational experience, but the test results indicate that the method is promising.

9 - CONCLUSIONS

We have shown how to incorporate an ellipsoid variable metric in a bundle method for convex minimization. The method seems to be promising. We hope to increase its efficiency by using more refined ellipsoid updates.

REFERENCES

- [A1] M. Akgül, Topics in Relaxation and Ellipsoidal Methods, Research Notes in Mathematics 97, Pitman, Boston, 1984.
- [B1] R.B. Bland, D. Goldfarb and M.J. Todd, The ellipsoid method : a survey, Operations Research, 29 (1981), pp.1039-1091.
- [D1] V.F. Demyanov and L.V. Vasilev, Nondifferentiable Optimization, Nauka, Moscow, 1981 (English translation : Optimization Software Inc./Springer, New York, 1985).

- [E1] A. Ech-Cherif and J.G. Ecker, A class of rank-two algorithms for convex programming, *Mathematical Programming*, 29 (1984), pp. 187-202.
- [E2] J.G. Ecker and M. Kupferschmid, An ellipsoid algorithm for nonlinear programming, *Mathematical Programming*, 27 (1983), pp. 1-15.
- [G1] V.I. Gershovich, On a cut-off method using linear space transformations, in : *Theory of Optimal Solution*, Institute of Cybernetics, Kiev, 1979, pp. 15-23 (in Russian).
- [G2] P.E. Gill, G.M Golub, W. Murray and M.A Saunders, Methods for modifying matrix factorizations, *Mathematics of Computation*, 28 (1974), pp. 505-535.
- [G3] P.E. Gill, W. Murray and M.A. Saunders, Methods for computing and modifying the LDV factors of a matrix, *Mathematics of Computation*, 29 (1975), pp. 1051-1077.
- [G4] J.L. GOFFIN, Variable metric relaxation methods, part I : A conceptual algorithm, Technical Report SOL 81-16, Systems Optimization Laboratory, Stanford University, 1981.
- [G5] J.L. Goffin, Convergence results in a class of variable metric subgradient methods, in : *Nonlinear Programming 4*, O.L. Mangasarian, R.R. Meyer and S.M. Robinson, eds., Academic Press, New York, 1981, pp. 283-326.
- [G6] J.L. Goffin, Convergence rates of the ellipsoid method on general convex functions, *Mathematics of Operations Research*, 8 (1983) pp. 135-150.
- [G7] J.L. Goffin, Variable metric relaxation methods, part II : the ellipsoid method, *Mathematical Programming*, 30 (1984), pp. 147-162.
- [G8] D. Goldfarb and M.J. Todd, Modifications and implementation of the ellipsoid algorithm for linear programming, *Mathematical Programming*, 23 (1982), pp. 1-19.

- [K1] J.E. Kelley, The cutting plane method for solving convex programs, Journal of SIAM, 8 (1960), pp. 703-712.
- [K2] K.C. Kiwiel, Methods of Descent for Nondifferentiable Optimization, Lecture Notes in Mathematics 1133, Springer, Berlin, 1985.
- [K3] K.C. Kiwiel, A method for solving certain quadratic programming problems arising in nonsmooth optimization, IMA Journal of Numerical Analysis, 6 (1986), pp. 137-152.
- [L1] C. Lemaréchal, Numerical experiments in nonsmooth optimization, in : Progress in Nondifferentiable Optimization, E.A. Nurminski, ed., CP-82-S8, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1982, pp. 61-84.
- [L2] C. Lemaréchal and R. Mifflin, eds., Nonsmooth Optimization, Pergamon Press, Oxford, 1978.
- [L3] C. Lemaréchal and J.J. Strodiot, Bundle methods, cutting plane algorithms and σ -Newton directions, in : Nondifferentiable Optimization : Motivations and Applications, V.F. Demyanov and D. Pallaschke ; eds., Lecture Notes in Economics and Mathematical Systems 255, Springer, Berlin, 1985, pp. 25-33.
- [L4] C. Lemaréchal and J. Zowe, Some remarks on the construction of higher order algorithms for convex optimization, Journal of Applied Mathematics and Optimization, 10 (1983), pp. 51-68.
- [M1] R. Mifflin, Better than linear convergence and safeguarding in nonsmooth minimization, in : System Modelling and Optimization, P. Thoft-Christensen, ed., Lecture Notes in Control and Information Sciences 59, Springer, Berlin, 1984, pp. 321-330.
- [M2] J.J. Moré, Recent developments in algorithms and software for trust region methods, in : Mathematical Programming, The State of the Art, Bonn 1982, A. Bachem, M. Grötschel and B. Korte, eds., Springer, Berlin, 1983, pp. 258-287.

- [S1] N.Z. Shor, Minimization Methods for Non-Differentiable Functions, Springer, Berlin, 1985.
- [S2] N.Z. Shor and V.I. Gershovich, Family of algorithms for solving convex programming problems, Kibernetika, n° 4 (1979), pp. 62-67 (in Russian ; English translation : Cybernetics, 15 (1980), pp. 502-508).
- [S3] G. Sonnevend, A modified ellipsoid method for the minimization of convex functions with superlinear convergence (or finite termination) for well-conditioned C^3 smooth (or piecewise linear) functions, in : Nondifferentiable Optimization : Motivations and Applications, V.F. Demyanov and D. Pallaschke, eds., Lecture Notes in Economics and Mathematical Systems 255, Springer, Berlin, 1985, pp. 264-277.
- [T1] M.J. Todd, On minimum volume ellipsoids containg part of a given ellipsoid, Mathematics of Operations Research, 7 (1982), pp. 253-261.
- [Y1] D.B. Yudin and A.S. Nemirovskii, Informational complexity and effective methods for the solution of convex extremal problems, Ekonomika i Matematicheskie Metody, 12 (1976), pp. 357-369 (in Russian ; English translation : MATEKON 13 (1977), pp. 3-25).
- [Y2] D.B. Yudin, A.P. Goriashko and A.S. Nemirovskii, Mathematical Optimization Methods for Devices and Algorithms of Automatic Control Systems, Radio i Sviaz, Moscow, 1982, (in Russian).
- [Z1] J. Zowe, Nondifferentiable optimization -a motivation and a short introduction into the subgradient- and the bundle concept, in : Computational Mathematical Programming, K. Schittkowski, ed., Springer, Berlin, 1985, pp. 323-356.

Table 8.1. Test results of a descent ellipsoid method

Problem	N	M _g	f(x*)	ϵ_s	f(x ¹)	δ	$ x^1 - x^* _{A_1}$	V ¹	k	f(x ^k)	V ^k	m _e
Shor	5	10	22.60016	1E-6	80	$\delta^{1,a}$	1	54.8	46	22.60016	6E-23	817
						$\delta^{1,b}$	0.51	9.4E+3	49	22.60016	4E-22	888
						$\delta^{1,c}$	0.10	2.9E+7	59	22.60016	8E-22	960
Colville 1	5	10	-32.3487	1E-5	20	$\delta^{2,a}$	1	3.92	47	-32.3486	2E-21	721
						$\delta^{2,b}$	0.48	294	52	-32.3486	5E-21	735
						$\delta^{2,c}$	0.05	2.9E+7	54	-32.3486	1E-21	961
Rosen-Susuki	4	10	-44	1E-5	0	$\delta^{3,a}$	0.87	1.6E-2	23	-43.9999	3E-11	194
						$\delta^{3,b}$	0.61	1.3E+3	34	-43.9999	1E-11	364
						$\delta^{3,c}$	0.12	7.9E+5	42	-43.9999	2E-11	440
Maxquad	10	20	-0.8414	1E-4	0	$\delta^{4,a}$	1	1.2E-7	79	-0.84138	1E-22	1294
						$\delta^{4,b}$	0.36	2.55	98	-0.84138	1E-21	1950
						$\delta^{4,c}$	0.06	2.6E+8	118	-0.84138	5E-22	2648
Mxhilb	30	45	0	1E-6	3.995	$\delta^{5,a}$	0.2	2.9E+38	16	3.5E-7	2E+18	3261
						$\delta^{5,b}$	0.2	2.0E+48	21	2.0E-7	2E+48	2210
L1hilb	30	45	0	1E-6	41.09	$\delta^{6,a}$	0.2	2.9E+38	17	1.7E-11	4E+14	4898
						$\delta^{6,b}$	0.2	2.0E+48	22	5.9E-7	2E+41	3240

Table 8.2. Problem SHELL DUAL

k	$f(x^k)$	NFEV	v^k	m_e
70	32.6369	153	2.0E-2	582
78	32.5482	178	5.7E-3	606
97	32.4408	237	2.3E-4	654
107	32.4045	268	9.8E-5	669
110	32.3929	277	7.4E-5	677
112	32.3843	283	6.4E-5	679
118	32.3748	306	3.7E-5	690
127	32.3648	345	8.1E-6	709
165	32.3548	496	5.9E-9	793
191	32.3538	600	8.0E-11	845

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

